

TITLE OF THE INVENTION

MAP DATA PROCESSING UNIT

5 BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a map data processing unit capable of updating map data without affecting the operation speed of its navigation function with minimizing the time of precluding the use of the navigation function.

Description of Related Art

A conventional vehicle-mounted navigation system or the like usually operates on map data stored in a recording medium such as a CD-ROM or DVD-ROM, and updates the map data by changing the CD-ROM or DVD-ROM. The method, however, cannot update the data frequently, only once or twice a year at most, and hence it cannot always provide the latest map data.

In contrast with this, a method is proposed of updating the map data in a shorter period of time by storing the map data in a rewritable medium (refer to Relevant Reference 1, for example).

Relevant Reference 1: Japanese patent application laid-open No. 2002-49624 (page 4, Fig. 7).

25 With such an arrangement, the conventional map data processing unit updates the map data when it becomes necessary. Thus, when using a plurality of map data in processing such as route search, it imposes a heavy load on the navigation system, thereby presenting a problem of retarding the operation.

30 In addition, carrying out the update processing of all the

map data at once offers a problem of precluding the use of the vehicle-mounted navigation because the update processing takes a lot of time. For example, to rewrite all the map data by reading data from a DVD-ROM, about nine gigabit data must be read. Since a current DVD-ROM drive usually reads about 1-2 MB per second, it takes about 1-2 hours to read the map data, with precluding the use of the navigation during that time.

SUMMARY OF THE INVENTION

The present invention is implemented to solve the foregoing problem. It is therefore an object of the present invention to provide a map data processing unit capable of updating the map data without affecting the operation speed of its navigation function with minimizing the time of disabling the navigation function.

According to a first aspect of the present invention, there is provided a map data processing unit that has a function of updating map data, and includes a display section for performing various types of display including display of a processing result of the map data, a manipulation section for carrying out input manipulation for processing the map data, and a control section for performing entire control, the map data processing unit including: an update data input section for inputting update data for updating the map data to latest version map data; a batch update section for collectively updating all the map data to the latest version map data in accordance with the update data fed from the update data input section; a dynamic update section for updating the map data to the latest version map data in accordance with the update data fed from the update data input section every time the map data becomes necessary; an update

method selecting section for selecting one of the batch update section and dynamic update section in response to the update data; and a map data storing section for storing the map data.

Thus, the map data processing unit can switch the update method between the batch update and the dynamic update in response to the update data. As a result, it offers an advantage of being able to perform the update to the latest version map data without affecting the operation speed of the navigation, with minimizing the time of precluding the use of the navigation.

According to a second aspect of the present invention, there is provided a map data processing unit that has a function of updating map data, and includes a display section for performing various types of display including display of a processing result of the map data, a manipulation section for carrying out input manipulation for processing the map data, and a control section for performing entire control, the map data processing unit including: a post-update data cache section for temporarily storing, in a cache, latest version map data to be read in response to a request for map data; an update data input section for inputting update data for updating the map data to the latest version map data, when the post-update data cache section does not store the map data; and a dynamic update section for updating, when the update data input section inputs the update data, old version map data to latest version map data to be stored in the cache, in accordance with the update data fed from the update data input section every time a request for the map data occurs.

Thus, the map data processing unit can utilize the latest version map data by only reading it from the cache using the post-update data cache section, if the map data associated with

the request from an application is stored in the cache by the post-update data cache section. As a result, it offers an advantage of being able to reduce the time of the dynamic update processing.

5 According to a third aspect of the present invention, there is provided a map data processing unit including: an update data input section for inputting update data for updating old version map data to latest version map data; an update data recording section for recording the update data input by the update data
10 input section; an update section for updating, when a map data request from an application occurs, the old version map data to the latest version map data in accordance with the old version map data and the update data stored in the update data recording section, which old version map data and update data correspond
15 to the map data associated with the map data request from the application; a map data storing section for storing the map data; a display section for displaying a processing result of the map data; a manipulation section for performing input manipulation for processing the map data; and a control section for carrying
20 out overall control.

 Thus, the map data processing unit offers an advantage of being able to use the update data input section for other purposes during the navigation operation by inputting from the update data input section the update data for updating the map data
25 associated with the request for the map data from the application at a timing different from the timing during the navigation operation, and by recording the update data into the update data recording section.

30 BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing a configuration of an embodiment 1 of a map data processing unit in accordance with the present invention;

5 Fig. 2 is a flowchart illustrating the operation of an update method selecting section of the embodiment 1 of the map data processing unit in accordance with the present invention;

Fig. 3 is a flowchart illustrating the operation of a batch update section of the embodiment 1 of the map data processing unit in accordance with the present invention;

10 Fig. 4 is a flowchart illustrating the operation of a dynamic update section of the embodiment 1 of the map data processing unit in accordance with the present invention;

Fig. 5 is a diagram illustrating a format of the map data stored in a map data storing section of the embodiment 1 of the map data processing unit in accordance with the present;

Fig. 6 is a diagram illustrating a format of the map data format stored in the map data storing section of the embodiment 1 of the map data processing unit in accordance with the present invention;

20 Fig. 7 is a diagram illustrating a format of local data of the embodiment 1 of the map data processing unit in accordance with the present invention;

Fig. 8 is a diagram illustrating a format of differential update data of the embodiment 1 of the map data processing unit in accordance with the present invention;

Fig. 9 is a diagram illustrating a format of update data of the embodiment 1 of the map data processing unit in accordance with the present invention;

30 Fig. 10 is a flowchart illustrating the operation of the batch update section of an embodiment 2 of the map data processing

unit in accordance with the present invention;

Fig. 11 is a flowchart illustrating the operation of the batch update section of an embodiment 3 of the map data processing unit in accordance with the present invention;

5 Fig. 12 is a flowchart illustrating the operation of the dynamic update section of the embodiment 3 of the map data processing unit in accordance with the present invention;

Fig. 13 is a block diagram showing a configuration of an embodiment 4 of the map data processing unit in accordance with
10 the present invention;

Fig. 14 is a flowchart illustrating the update operation of the map data by a differential data update section in the embodiment 4 of the map data processing unit in accordance with the present invention;

15 Fig. 15 is a flowchart illustrating the update operation of the map data by the differential data update section in an embodiment 5 of the map data processing unit in accordance with the present invention;

Fig. 16 is a block diagram showing a configuration of an
20 embodiment 6 of the map data processing unit in accordance with the present invention;

Fig. 17 is a flowchart illustrating the update operation of the map data by a post-update data cache section in the embodiment 6 of the map data processing unit in accordance with
25 the present invention;

Fig. 18 is a block diagram showing a configuration of an embodiment 7 of the map data processing unit in accordance with the present invention;

Fig. 19 is a block diagram showing a configuration of an
30 embodiment 8 of the map data processing unit in accordance with

the present invention;

Fig. 20 is a block diagram showing a configuration of an embodiment 9 of the map data processing unit in accordance with the present invention;

5 Fig. 21 is a flowchart illustrating the update operation of the map data by a current position acquisition section of the embodiment 9 of the map data processing unit in accordance with the present invention;

10 Fig. 22 is a block diagram showing a configuration of an embodiment 10 of the map data processing unit in accordance with the present invention;

Fig. 23 is a flowchart illustrating the update operation of the map data by the current position acquisition section and pre-update section in the embodiment 10 of the map data
15 processing unit in accordance with the present invention;

Fig. 24 is a block diagram showing a configuration of an embodiment 11 of the map data processing unit in accordance with the present invention;

20 Fig. 25 is a flowchart illustrating the update operation of the map data using a location registration section and post-update data cache section of the embodiment 11 of the map data processing unit in accordance with the present invention;

25 Fig. 26 is a block diagram showing a configuration of an embodiment 12 of the map data processing unit in accordance with the present invention;

Fig. 27 is a flowchart illustrating the update operation of the map data using the location registration section and pre-update data recording section of the embodiment 12 of the map data processing unit in accordance with the present
30 invention;

Fig. 28 is a block diagram showing a configuration of an embodiment 13 of the map data processing unit in accordance with the present invention;

5 Fig. 29 is a flowchart illustrating the update operation of the map data using a driving history storing section and post-update data cache section of the embodiment 13 of the map data processing unit in accordance with the present invention;

Fig. 30 is a block diagram showing a configuration of an embodiment 14 of the map data processing unit in accordance with
10 the present invention;

Fig. 31 is a flowchart illustrating the update operation of the map data using the driving history storing section and pre-update section of the embodiment 14 of the map data processing unit in accordance with the present invention;

15 Fig. 32 is a block diagram showing a configuration of an embodiment 15 of the map data processing unit in accordance with the present invention;

Fig. 33 is a flowchart illustrating the update data recording operation into the update data recording section of
20 the embodiment 15 of the map data processing unit in accordance with the present invention; and

Fig. 34 is a flowchart illustrating the update operation of the map data by the dynamic update section of the embodiment 15 of the map data processing unit in accordance with the present
25 invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The invention will now be described with reference to the accompanying drawings.

30 EMBODIMENT 1

Fig. 1 is a block diagram showing a configuration of an embodiment 1 of the map data processing unit. It comprises a map data storing section 101 for storing map data; an update data input section 102 for inputting update data; a batch update section 103 for collectively updating the map data stored in the map data storing section 101 using one or more update data input via the update data input section 102; a dynamic update section 104 for updating the map data stored in the map data storing section 101 using one or more update data input via the update data input section 102 when necessary; an update method selecting section 105 for selecting one of the batch update section 103 and dynamic update section 104 in response to differential data; a display section 106 for carrying out display based on the map data; a manipulation section 107 for carrying out various manipulations; and a control section 108 for controlling the entire system.

The map data storing section 101 for storing the map data consists of a rewritable medium such as a hard disk for storing the map data after update. The update data input section 102 for inputting the update data consists of a data storing medium such as a CD-ROM, DVD-ROM or memory card. The batch update section 103 collectively updates the map data stored in the map data storing section 101 using one or more update data supplied from the update data input section 102. The dynamic update section 104 updates the map data stored in the map data storing section 101 using one or more update data supplied from the update data input section 102 as needed. The update method selecting section 105 selects one of the batch update section 103 and dynamic update section 104 in response to the update data. The display section 106 carries out the display in accordance with

the map data.

Fig. 5 is a diagram illustrating a format of the map data stored in the map data storing section 101; and Fig. 6 is a diagram illustrating a format of a map data file stored in the map data storing section 101.

The map data consists of files, each of which includes one type of data. Each file contains header information and a plurality of local data (data #0 - data #n). The header information consists of a data type, data version, file size and the number of the data as illustrated in Fig. 6. The data type indicates whether the data is used for map display, for location, for route search or for route guidance corresponding to the individual functions of the navigation system.

The local data refers to data on individual regions obtained by dividing the entire area. For example, data on whole Japan as illustrated in Fig. 5 correspond to the entire map data file as illustrated in Fig. 6, and the individual areas indicated by a mesh in Fig. 5 correspond to local data (local data #0 - #n).

Fig. 7 is a diagram illustrating formats of the local data. Fig. 7(a) illustrates a format of the local data used for map display. It includes the data size of the local data; the data range of the local data; and data actually used for the map display such as road line coordinate strings representing geometry of roads, background line coordinate strings representing geometry of rivers, railroads and the like, display character strings for displaying place names and the like, and display symbols for representing entities such as a school and hospital. Fig. 7(b) illustrates a format of the local data for route search. It includes the data size of the local data; the

data range of the local data; node data mainly representing intersections; link data representing the road data, and traffic regulation data.

As for the update data input via the update data input section 102 as shown in Fig. 1, there are two types of data: a complete set of update data for replacing the old map data; and differential update data for carrying out the update in accordance with the difference between the old version and latest version of the map data. The format of the complete set of update data is the same as that of the map data illustrated in Figs. 5 and 6.

Fig. 8 is a diagram illustrating a format of the differential update data. Just as the map data, the differential update data consists of files, each of which includes one type of data. Each file contains header information and a plurality of update data #0 - #n. The update data #0 - #n correspond to the local data #0 - #n as illustrated in Fig. 6.

The header information consists of information items indicating a data type, a corresponding data version, a post-update data version, a file size and the number of the update data. The data type corresponds to the data type of the map data of Fig. 6. The corresponding data version indicates the version of the map data to be updated. For example, when the corresponding data version is version 1.0, and the post-update data version is version 1.1, the differential update data is the data for updating the map data of version 1.0 to the map data of version 1.1.

Fig. 9 is a diagram illustrating a format of the update data consisting of a data size indicating the size of the update

data itself, a post-update data size indicating the size of the map data after the update, and the body of the update data for carrying out the map update.

Next, the operation of the present embodiment 1 will be
5 described.

Fig. 2 is a flowchart illustrating the operation of the update method selecting section 105 in the embodiment 1 of the map data processing unit. Fig. 3 is a flowchart illustrating the operation of the batch update section 103, and Fig. 4 is
10 a flowchart illustrating the operation of the dynamic update section 104.

First, the operation of the update method selecting section 105 will be described with reference to the flowchart of Fig. 2. The update method selecting section 105 acquires header
15 information of the update data from the update data input section 102 at step ST1201. Subsequently, the update method selecting section 105 decides the data type of the header information at step ST1202. If it is the data for route search, the update method selecting section 105 has the batch update section 103
20 carry out the batch update at step ST1203. If the data is not the data for route search, the update method selecting section 105 registers the data as the data to be subjected to the dynamic update at step ST1204. Finally, the update method selecting section 105 makes a decision as to whether the header information
25 of all the data has been input at step ST1205, and terminates the processing if the input has been completed. Otherwise, the update method selecting section 105 repeats the processing from step ST1201.

Next, the operation of the batch update section 103 will
30 be described with reference to Fig. 3.

The batch update section 103 deletes the map data to be updated at step ST1401, and records the corresponding complete set of update data in the map data storing section 101 at step ST1402. Then, the batch update section 103 changes the file name of the complete set of update data to that of the deleted map data at step ST1403.

Next, the operation of the dynamic update section 104 will be described with reference to Fig. 4.

In response to a request for map data from the application at step ST1301, the dynamic update section 104 reads the map data from the map data storing section 101 at step ST1302. Then, the dynamic update section 104 makes a decision as to whether the data to be subjected to the dynamic update corresponding to the read map data is present or not at step ST1303. If the data is not present, the dynamic update section 104 terminates the processing. Otherwise, the dynamic update section 104 acquires the corresponding update data at step ST1304, and updates the map at step ST1305.

Thus, the batch update section 103 updates the set of data for route search to the latest version map data, and records the latest version map data in the map data storing section 101. On the other hand, the dynamic update section 104 updates each of the remaining data every time the application requires it.

The processing time of Fig. 3 is nearly the read time of the complete set of update data, which corresponds to the size of the map data for route search of about several tens of megabytes. Thus, the processing can be completed in several tens of seconds in the case of reading the data from the DVD-ROM.

Although the foregoing description is made by way of example in which the data for route search is subjected to the

batch update processing, data other than the route search data, which are considered to be rather small in size, can be used as the data to be subjected to the batch update processing.

5 In addition, although the target data to be subjected to the batch update processing is decided according to the data type, this is not essential. For example, the target data to be subjected to the batch update processing can be decided according to the data size rather than the data type.

Furthermore, although the update data input section 102
10 is assumed to be a data storing medium, this is not essential. For example, the update data can be input via various types of communication means such as a mobile phone, DSRC (Dedicated Short Range Communication), Bluetooth, and wireless LAN.

As described above, the present embodiment 1 makes a
15 decision as to the processing time and storage capacity required for carrying out the batch update processing from the map data type, map data size and the like; selects the data to be subjected the batch update; carries out the batch update processing; and carries out the dynamic update processing of the remaining data
20 as needed. Thus, the present embodiment 1 offers an advantage of being able to provide a map data processing unit capable of updating to the latest map data without affecting the operation speed of the navigation function, with minimizing the time of disabling the navigation function.

25

EMBODIMENT 2

Fig. 10 is a flowchart illustrating the operation of the batch update section of the present embodiment 2 of the map data processing unit. The configuration of the map data processing
30 unit to which the batch update section of the present embodiment

2 is applied is the same as that of Fig. 1.

Referring to Fig. 10, the operation of the batch update section of the map data processing unit will be described.

The batch update section moves the map data to be updated to another directory to keep the map data before the update at step ST1501. It records the corresponding complete set of update data into the map data storing section 101 at step ST1502, and changes the file name of the complete set of update data at step ST1503.

Thus, the present embodiment 2 keeps the map data before the update. Accordingly, it offers an advantage of being able to recover the state before the update with ease, and hence to provide the map data processing unit capable of carrying out the navigation using the map data of the previous version.

EMBODIMENT 3

Fig. 11 is a flowchart illustrating the operation of the batch update section of the present embodiment 3 of the map data processing unit; and Fig. 12 is a flowchart illustrating the operation of its dynamic update section. The configuration of the map data processing unit to which the batch update section and dynamic update section of the present embodiment 3 are applied is the same as that of Fig. 1.

Referring to the flowchart of Figs. 11 and 12, the operation of the present embodiment 3 will be described.

First, the operation of the batch update section 103 will be described with reference to Fig. 11. The batch update section 103 clears a batch update completion flag at step ST3001 to indicate that the batch update has not yet been completed. Then, the batch update section 103 records the corresponding complete

set of update data in the map data storing section 101 at step ST3002, and displays a batch update preparation completion telop on the display section 106 shown in Fig. 1 at step ST3003. Subsequently, the batch update section 103 deletes the map data at step ST3004, and changes the file name of the complete set of update data for the batch update to the file name of the deleted map data at step ST3005. Finally, the batch update section 103 sets the batch update completion flag at step ST3006 to indicate that the batch update has been completed.

10 Next, the operation of the dynamic update section 104 will be described with reference to Fig. 12.

In response to a map data request from an application at step ST3101, the dynamic update section 104 reads the appropriate map data from the map data storing section 101 at step ST3102. Then, the dynamic update section 104 checks the batch update completion flag to decide as to whether the batch update completion flag has been cleared or not at step ST3103. If it has been cleared, and hence indicates ~~that the batch update~~ has not yet been completed, the dynamic update section 104 terminates the dynamic update processing. As a result, the navigation is available using the old version map data read from the map data storing section 101 in response to the map data request from the application even during the batch update. On the other hand, if the batch update completion flag has been set, and hence indicates that the batch update has been completed, the dynamic update section 104 makes a decision as to whether the data to be subjected to the dynamic update corresponding to the map data read from the map data storing section 101 is present or not at step ST3104. If it is not present, the dynamic update section 104 terminates the dynamic update processing.

In contrast, if it is present, the dynamic update section 104 acquires the corresponding update data at step ST3105, and updates the map in accordance with the dynamic update processing at step ST3106.

5 As described above, the present embodiment 3 can read, in response to the map data request from the application, the old version map data from the map data storing section 101 even during the batch update processing that continues until the batch update is completed and the batch update completion flag is set,
10 thereby making available the navigation using the old version map data. Thus, the present embodiment 3 offers an advantage of being able to provide the map data processing unit capable of increasing the convenience of users.

15 EMBODIMENT 4

 Fig. 13 is a block diagram showing a configuration of the present embodiment 4 of the map data processing unit. In Fig. 13, the same or like portions to those of Fig. 1 are designated by the same reference numerals, and their description is omitted
20 here.

 In the present embodiment 4 of the map data processing unit, the batch update section 103 comprises a differential data update section 201 and an entire data update section 202.

 The differential data update section 201 carries out the
25 map update in accordance with the differential data between the old version and latest version map data. In contrast, the entire data update section 202 carries out the map update by rewriting the old complete set of map data to the latest version map data in accordance with the corresponding new complete set of update
30 data.

Fig. 14 is a flowchart illustrating the update operation of the map data by the differential data update section 201. On the other hand, the flow of the update operation of the map data by the entire data update section 202 is given by Fig. 3 described in connection with the foregoing embodiment 1.

Referring to Fig. 14, the map update operation by the differential data update section 201 will be described.

First, the differential data update section 201 reads the old version map data recorded in the map data storing section 101 at step ST1601. Second, the differential data update section 201 acquires the corresponding differential update data from the update data input section 102 at step ST1602, and updates the map in accordance with the old version map data and differential update data at step ST1603. Then, the differential data update section 201 records the post-update map data in the map data storing section 101 at step ST1604.

Subsequently, the differential data update section 201 makes a decision as to whether all the data have been updated or not at step ST1605. If the map data that has not yet been updated is left, the differential data update section 201 repeats the processing from step ST1601. In contrast, if it completes the update of all the map data, the differential data update section 201 deletes the map data before the update at step ST1606. Then, the differential data update section 201 changes the file name of the map data after the update to that of the map data before the update at step ST1607.

As described above, the present embodiment 4 enables the batch update using the complete set of update data or the differential update data. Thus, it offers an advantage of being able to provide the map data processing unit capable of reducing

the amount of the update data.

EMBODIMENT 5

In the present embodiment 5 of the map data processing unit,
5 the batch update section 103 has also the differential data
update section and entire data update section as in the
embodiment 4 of the map data processing unit. The entire data
update section updates the map data just as that of the foregoing
embodiment 4. However, the differential data update section
10 updates the map data in a different manner. Fig. 15 is a
flowchart illustrating the update operation of the map data by
the differential data update section.

Referring to Fig. 15, the update operation of the map data
by the differential data update section of the present
15 embodiment 5 will be described.

The differential data update section reads the map data
recorded in the map data storing section 101 at step ST1701,
and acquires the differential update data from the update data
input section 102 at step ST1702. Then, the differential data
20 update section updates the map at step ST1703, and records the
map data after the update in the map data storing section 101
at step ST1704. Subsequently, the differential data update
section makes a decision as to whether it updates all the data
at step ST1705. If the map data that has not yet been updated
25 is left, the differential data update section repeats the
processing from step ST1701. If the update of all the map data
has been completed, the differential data update section moves
the map data before the update to another directory at step ST1706,
and changes the file name of the map data after the update at
30 step ST1707.

Thus, the present embodiment 5 stores the map data before the update to the separate directory. Accordingly, it offers an advantage of being able to provide the map data processing unit capable of recovering the state before the update with ease.

5 In addition, the present embodiment 5 can carry out the batch update by using either the complete set of update data or the differential update data. Thus, it offers an advantage of being able to provide the map data processing unit capable of reducing the amount of the update data by using the
10 differential update data.

EMBODIMENT 6

Fig. 16 is a block diagram showing a configuration of the present embodiment 6 of the map data processing unit. In Fig.
15 16, the same or like portions to those of Fig. 1 are designated by the same reference numerals, and their description is omitted here. The present embodiment 6 of the map data processing unit comprises a post-update data cache section 301 for caching the map data updated by the dynamic update section 104, and a deletion
20 data decision section 302 for deciding map data to be deleted among the map data cached.

Fig. 17 is a flowchart illustrating the update operation of the map data using the post-update data cache section 301 in the embodiment 6 of the map data processing unit.

25 Referring to the flowchart of Fig. 17, the operation will be described.

When a request for map data is delivered from an application at step ST1801, the control section 108 makes a decision as to whether the requested map data is present in the cache or not
30 at step ST1802. If the map data is present in the cache, the

control section 108 reads the appropriate map data from the cache at step ST1803. Then, after updating the access history of the cache at step ST1804, the control section 108 terminates the processing. In contrast, if the map data is not present in the
5 cache, the control section 108 reads the appropriate map data from the map data storing section 101 at step ST1805, and acquires the corresponding differential update data from the update data input section 102 at step ST1806. Then, the control section 108 has the dynamic update section 104 update the map at step ST1807.
10 Subsequently, the control section 108 makes a decision as to whether the cache has available space or not at step ST1808. If it has no available space, the control section 108 deletes the oldest map data in the access history from the cache at step ST1809. In contrast, if the cache includes available space,
15 or after deleting the oldest map data in the access history from the cache, the control section 108 writes the updated map data into the cache at step ST1810. Finally, after setting the access history at step ST1811, the control section 108 terminates the processing.

20 As described above, the present embodiment 6 includes the post-update data cache section 301. Accordingly, if the post-update data cache section 301 includes the map data requested from the application, the present embodiment 6 can utilize the map data of the latest version by only reading the
25 map data by the post-update data cache section 301.

Consequently, the present embodiment 6 offers an advantage of being able to provide the map data processing unit capable of reducing the time of the dynamic update processing.

30 EMBODIMENT 7

Fig. 18 is a block diagram showing a configuration of the present embodiment 7 of the map data processing unit. In Fig. 18, the same or like portions to those of Fig. 16 are designated by the same reference numerals, and their description is omitted here. The present embodiment 7 of the map data processing unit comprises a pre-update section 401 for carrying out map update beforehand, and a pre-update data decision section 402 for deciding the map data to be updated by the pre-update section 401 beforehand.

According to the present embodiment 7, the pre-update data decision section 402 decides the map data to be updated by the pre-update section 401 beforehand, and the pre-update section 401 carries out the map update beforehand and stores the map data into the post-update data cache in advance. Accordingly, the present embodiment 7 offers an advantage of being able to provide a map data processing unit capable of reducing the time of the dynamic update processing of the map data used at a high frequency by storing the highly used map data in the post-update data cache.

EMBODIMENT 8

Fig. 19 is a block diagram showing a configuration of the present embodiment 8 of the map data processing unit. In Fig. 19, the same or like portions to those of Fig. 18 are designated by the same reference numerals, and their description is omitted here. The present embodiment 8 of the map data processing unit comprises a pre-update data recording section 501 for recording the map data updated by the pre-update section 401 in advance. According to the present embodiment 8, the pre-update data recording section 501 stores the map data updated in advance.

Consequently, if the stored map data is the map data used at high frequency, the present embodiment 8 offers an advantage of being able to provide a map data processing unit capable of reducing the time of the dynamic update processing of the highly used map data.

EMBODIMENT 9

Fig. 20 is a block diagram showing a configuration of the present embodiment 9 of the map data processing unit. In Fig. 20, the same or like portions to those of Fig. 18 are designated by the same reference numerals, and their description is omitted here. The present embodiment 9 of the map data processing unit comprises a current position acquisition section 601 for detecting the position of a vehicle it belongs to.

Fig. 21 is a flowchart illustrating the update operation of the map data by using the current position acquisition section 601 in the embodiment 9 of the map data processing unit.

Referring to the flowchart of Fig. 21, the operation will be described.

First, the pre-update section 401 acquires the current position from the current position acquisition section 601 at step ST1901. Second, the pre-update section 401 reads the map data around the current position from the map data storing section 101 at step ST1902, and acquires the differential update data corresponding to the map data from the update data input section 102 at step ST1903. Then, the pre-update section 401 updates the map at step ST1904. Subsequently, the pre-update section 401 has the post-update data cache section 301 write the updated map data into the cache at step ST1905, and sets the access history at step ST1906. Finally, the pre-update

section 401 makes a decision as to whether all the surrounding maps have been read at step ST1907, and terminates the processing if all the maps have been read. If there is some map data left, the pre-update section 401 repeats the processing from step
5 ST1902.

According to the present embodiment 9, the post-update data cache stores the update data around the position of the vehicle (current position). Thus, the present embodiment 9 offers an advantage of being able to provide a map data processing unit
10 capable of reducing the time of the dynamic update processing of the map data around the position of the vehicle, which is used at high frequency.

EMBODIMENT 10

15 Fig. 22 is a block diagram showing a configuration of the present embodiment 10 of the map data processing unit. In Fig. 22, the same or like portions to those of Fig. 20 are designated by the same reference numerals, and their description is omitted here.

20 The present embodiment 10 of the map data processing unit comprises a pre-update data recording section 501. The pre-update data recording section 501 stores the map data updated by the pre-update section 401 in advance.

25 Fig. 23 is a flowchart illustrating the update operation of the map data by using the current position acquisition section 601 and pre-update section 401 in the embodiment 10 of the map data processing unit.

Referring to the flowchart of Fig. 23, the operation will be described.

30 First, the pre-update section 401 acquires the current

position from the current position acquisition section 601 at step ST2001. Subsequently, the pre-update section 401 reads the map data around the current position from the map data storing section 101 at step ST2002, and acquires the corresponding differential update data from the update data input section 102 at step ST2003, followed by updating the map at step ST2004. Then, the pre-update section 401 writes the map data updated in advance into the pre-update data recording section 501 at step ST2005, and makes a decision as to whether all the surrounding maps have been read at step ST2006. If all the surrounding maps have been read, the pre-update section 401 terminates the processing. In contrast, when the map data that has not yet been read is left, the pre-update section 401 repeats the processing from step ST2002.

According to the present embodiment 10, the pre-update data recording section 501 stores the map data updated in advance by the pre-update section 401. Thus, the present embodiment 10 offers an advantage of being able to provide a map data processing unit capable of reducing the time of the dynamic update processing of the map data around the position of the vehicle, which is used at high frequency.

EMBODIMENT 11

Fig. 24 is a block diagram showing a configuration of the present embodiment 11 of the map data processing unit. In Fig. 24, the same or like portions to those of Fig. 18 are designated by the same reference numerals, and their description is omitted here.

The present embodiment 11 of the map data processing unit comprises a location registration section 801 for carrying out

location registration.

Next, the operation of the present embodiment 11 will be described.

Fig. 25 is a flowchart illustrating the update operation
5 of the map data using the location registration section 801 and post-update data cache section 301 of the embodiment 11 of the map data processing unit. Referring to the flowchart of Fig. 25, the operation will be described.

First, the pre-update section 401 acquires a registration
10 position from the location registration section 801 at step ST2101.

Subsequently, the pre-update section 401 reads the map data around the registration position from the map data storing section 101 at step ST2102, and acquires the corresponding
15 differential update data from the update data input section 102 at step ST2103, followed by updating the map at step ST2104.

Then, the pre-update section 401 has the post-update data cache section 301 write the updated map data into the cache at step ST2105, and sets the access history at step ST2106.
20 Subsequently, the pre-update section 401 makes a decision as to whether all the surrounding maps have been read or not at step ST2107. If all the surrounding maps have been read, the pre-update section 401 terminates the processing. If there is some map data left, the pre-update section 401 repeats the
25 processing from step ST2102.

According to the present embodiment 11, the post-update data cache stores the update data around the registration position. Thus, the present embodiment 11 offers an advantage of being able to provide a map data processing unit capable of
30 reducing the time of the dynamic update processing of the map

data around the registration location, which is used at high frequency.

EMBODIMENT 12

5 Fig. 26 is a block diagram showing a configuration of the present embodiment 12 of the map data processing unit. In Fig. 26, the same or like portions to those of Fig. 24 are designated by the same reference numerals, and their description is omitted here.

10 The present embodiment 12 of the map data processing unit comprises a pre-update data recording section 501 which is not installed in the foregoing embodiment 11. The pre-update data recording section 501 stores the map data updated by the pre-update section 401 in advance as described above in the
15 foregoing embodiment 10.

 Fig. 27 is a flowchart illustrating the update operation of the map data by using the location registration section 801 and pre-update data recording section 501 in the embodiment 12 of the map data processing unit. Referring to the flowchart
20 of Fig. 27, the operation will be described.

 First, the pre-update section 401 acquires the registration position from the location registration section 801 at step ST2201. Subsequently, the pre-update section 401 reads the map data around registration position from the map
25 data storing section 101 at step ST2202, and acquires the corresponding differential update data from the update data input section 102 at step ST2203, followed by updating the map at step ST2204. Then, the pre-update section 401 writes the updated map data into the pre-update data recording section 501
30 at step ST2205, and makes a decision as to whether all the

surrounding maps have been read at step ST2206. If all the surrounding maps have been read, the pre-update section 401 terminates the processing. In contrast, when the map data that has not yet been read is left, the pre-update section 401 repeats
5 the processing from step ST2202.

According to the present embodiment 12, the pre-update data recording section 501 stores the map data around the registration position. Thus, the present embodiment 12 offers an advantage of being able to provide a map data processing unit
10 capable of reducing the time of the dynamic update processing of the map data around the registration location which is used at high frequency.

EMBODIMENT 13

15 Fig. 28 is a block diagram showing a configuration of the present embodiment 13 of the map data processing unit. In Fig. 28, the same or like portions to those of Fig. 18 are designated by the same reference numerals, and their description is omitted here. The present embodiment 28 of the map data processing unit
20 comprises a driving history storing section 1001 for storing driving history.

Next, the operation of the present embodiment 13 will be described.

Fig. 29 is a flowchart illustrating the update operation
25 of the map data by using the driving history storing section 1001 and post-update data cache section 301 in the embodiment 13 of the map data processing unit. Referring to the flowchart of Fig. 29, the operation will be described.

First, the pre-update section 401 acquires the driving
30 history position from the driving history storing section 1001

at step ST2301. Then, the pre-update section 401 reads the map data around the driving history position from the map data storing section 101 at step ST2302, and acquires the corresponding differential update data from the update data input section 102 at step ST2303, followed by updating the map at step ST2304. Subsequently, the pre-update section 401 has the post-update data cache section 301 write the updated map data into the cache at step ST2305, and sets the access history at step ST2306. Finally, the pre-update section 401 makes a decision as to whether all the surrounding maps have been read or not at step ST2307. If all the maps have been read, the pre-update section 401 terminates the processing. If there is some map data left, the pre-update section 401 repeats the processing from step ST2302.

According to the present embodiment 13, the post-update data cache stores the updated map data around the driving history position. Thus, the present embodiment 13 offers an advantage of being able to provide a map data processing unit capable of reducing the time of the dynamic update processing of the map data around the driving history position which is used at high frequency.

EMBODIMENT 14

Fig. 30 is a block diagram showing a configuration of the present embodiment 14 of the map data processing unit. The present embodiment 14 of map data processing unit comprises the pre-update data recording section 501 the foregoing embodiment 13 of the map data processing unit lacks. As described above in connection with the foregoing embodiment 12, the pre-update data recording section 501 records the map data updated by the

pre-update section 401 in advance. In Fig. 30, since the remaining configuration is the same as that of the foregoing embodiment 13 of map data processing unit, their description is omitted here by designating the same or like portions to those of Fig. 28 by the same reference numerals.

Next, the operation of the present embodiment 14 will be described.

Fig. 31 is a flowchart illustrating the update operation of the map data by using the driving history storing section 1001 and pre-update section 401 in the present embodiment 14 of the map data processing unit. Referring to the flowchart of Fig. 31, the operation will be described.

First, the pre-update section 401 acquires the driving history position from the driving history storing section 1001 at step ST2401. Second, the pre-update section 401 reads the map data around the driving history position from the map data storing section 101 at step ST2402, acquires the corresponding differential update data from the update data input section 102 at step ST2403, and updates the map at step ST2404. Then, the pre-update section 401 writes the updated map data into the pre-update data recording section 501 at step ST2405. Finally, the pre-update section 401 makes a decision as to whether all the surrounding maps have been read at step ST2406. If all the maps have been read, the pre-update section 401 terminates the processing. If there is some map data left, the pre-update section 401 repeats the processing from step ST2402.

Incidentally, as the driving history, it is possible to use not only coordinates of places passed by in actual drives, but also coordinates of destinations or places on a route which are set in route search, or coordinates of places of a search

route.

As described above, according to the present embodiment 14, the pre-update data recording section 501 stores the map data updated by using the map data around the driving history position and the corresponding differential update data. Thus, the present embodiment 14 offers an advantage of being able to provide a map data processing unit capable of reducing the time of the dynamic update processing of the map data around the driving history position, which is used at high frequency.

10

EMBODIMENT 15

Fig. 32 is a block diagram showing a configuration of the present embodiment 15 of the map data processing unit. In Fig. 32, the same or like portions to those of Fig. 1 are designated by the same reference numerals, and their description is omitted here. The present embodiment 15 of the map data processing unit comprises an update data recording section 3201 for recording the update data fed from the update data input section 102.

Next, the operation of the present embodiment 15 will be described.

20

Fig. 33 is a flowchart illustrating the update data recording operation into the update data recording section 3201 of the embodiment 15, and Fig. 34 is a flowchart illustrating the update operation of the map data by the dynamic update section 104 of the embodiment 15.

25

Referring to the flowcharts of Figs. 33 and 34, the operation will be described.

First, the recording operation of the update data into the update data recording section 3201 will be described with reference to the flowchart of Fig. 33. The control section 108

30

acquires the update data from the update data input section 102 at a timing different from the timing of a navigation operation at step ST3301, and records the update data in the update data recording section 3201 at step ST3302. Subsequently, the
5 control section 108 makes a decision as to whether all the update data have been input or not at step ST3303. If the input has been completed, the control section 108 terminates the processing, and otherwise it repeats the processing from step ST3301.

10 Next, the operation of the dynamic update section 104 will be described with reference to the flowchart of Fig. 34. In response to a request for map data from an application at step ST3401, the dynamic update section 104 reads the appropriate map data from the map data storing section 101 at step ST3402.
15 Then, the dynamic update section 104 reads the update data corresponding to the read map data from the update data recording section 3201 at step ST3403, followed by updating the map at step ST3404.

Thus, the present embodiment 15 updates the map in
20 accordance with the update data recorded in the update data recording section 3201. Therefore it is not necessary to input the update data from the update data input section 102 during the navigation operation. This makes it possible to use the update data input section 102 for other purposes during the
25 navigation operation. For example, when the update data input section 102 is a DVD, it can carry out DVD video replay during the navigation operation. When the update data input section 102 is a memory card, it is applicable to other navigation data exchange.

30 Furthermore, the speed-up of the map update itself can be

expected by using a high-speed medium such as a hard disk as the update data recording section 3201.

Although the foregoing description is made assuming that the update data recording section 3201 is installed separately
5 from the map data storing section 101, they may be incorporated into the same medium. For example, the same hard disk can record the map data and the update data.

As described above, the present embodiment 15 offers an advantage of being able to provide a map data processing unit
10 capable of using the update data input section 102 for other purposes during the navigation operation by recording the update data into the update data recording section 3201 at a timing different from the navigation operation period.